Baze de date

Limbajul SQL

Teams: FI-AIA-2-Baze de date-2022/2023

Adrian Runceanu

Curs 1

Noţiuni introductive despre teoria generală a bazelor de date

Câteva precizări

Structura cursului

- 2 ore curs – Sef lucr. dr. Adrian Runceanu
- 2 ore laborator – Sef lucr. dr. Adrian Runceanu

Forme de examinare

A) Examen final = 60%
B) Evaluare pe parcursul semestrului a activităţii de laborator = 40%

In timpul semestrului vor fi 2 teste, iar in ultima saptamana va fi evaluarea finala a activitatii de laborator
Testul 1 – SQL (notiuni generale) = 20 puncte
Testul 2 – SQL = 30 puncte
Testul final = 50 puncte
Total puncte posibil obtinute la laborator = 100 puncte
Dintre acestea se considera 40% pentru nota finala la disciplina Baze de date

Bibliografia necesară cursului:

1. **Dezvoltarea bazelor de date în Oracle 9i prin exemple**, Dan Hotka, Editura All, 2002.
2. **An Introduction to Database Systems**, C. J. Date, Addison-Wesley Publishing Company, New York, 1995.
3. **Database Management Systems**, Ramakrishnan, R., New York: McGraw-Hill, 1998.
4. **SQL. Dialecte DB2, Oracle, Visual FoxPro**, M. Fotache, ed. Polirom, 2001.
5. **Baze de date – Visual Foxpro 6.0 – îndrumar de laborator**, Marian Popescu, Adrian Runceanu, Editura Academica Brâncuşi, Târgu-Jiu, 2007.
6. **Baze de date – o abordare Visual Foxpro**, Adrian Runceanu, Mihaela Runceanu, Editura Academica Brâncusi, Târgu-Jiu, 2009

Bibliografia necesară cursului:

Referinţele bibliografice nr. 5 şi nr. 6 se pot împrumuta de la Biblioteca Facultăţii de Inginerie, Str. Geneva nr.3, Etaj I – lângă Decanat.

1. Suport curs - varianta electronică disponibilă pe site-ul: https://www.runceanu.ro/adrian

2. Îndrumar de laborator - varianta electronică disponibilă pe site pentru fiecare lucrare de laborator.

Notă: Actualizarea site-ului se face săptămânal.

Resurse Baze de date

1. Suport curs - varianta electronică disponibilă pe: https://www.runceanu.ro/adrian/

Notă: Actualizarea site-ului se face saptamanal.

2. curs pe Teams (FI-AIA-2-Baze de date-2022-2023)

3. laborator pe Teams (FI-AIA-2-Baze de date-2022-2023)

Resurse Baze de date

Resurse electronice pentru activitatile de laborator:

La fiecare laborator se va utiliza platforma APEX (Application Express ORACLE).

Datele de conectare:

https://apex.oracle.com/pls/apex

Workspace: **oracle_studenti_aia**

username: **numele e-mail-ului student(a)**

pass: **Student_2023** (se va schimba dupa prima conectare)

Resurse Baze de date

Tutoriale **APEX ORACLE**:

- https://www.youtube.com/watch?v=ahlW-Jkumto&list=PL_c9BZzLwBRJXk2hEP-U_OUMJAf_TYEiZ
- https://www.youtube.com/watch?v=PgqqNwjZ3o0&list=PL_c9BZzLwBRJXk2hEP-U_OUMJAf_TYEiZ
- https://www.youtube.com/watch?v=UwCpB2ElxbE&list=PL_c9BZzLwBRJXk2hEP-U_OUMJAf_TYEiZ&index=6

Introducere

Cursul de **Baze de date** adresează studenţilor înscrişi la programul de studii Automatică şi Informatică Aplicată, organizat de facultatea de Inginerie, anul 2, semestrul 2.

Obiectivul general al disciplinei al cursului de **Baze de date** vizează pregătirea studenţilor în domeniul utilizării bazelor de date în diferite domenii tehnice.

Se va avea in vedere insuşirea noţiunilor care stau la baza analizei, proiectării şi implementării unei aplicaţii cu baze de date.

Introducere

Obiectivele specifice:

Curs:

- ✓ Cunoaşterea noţiunilor privind bazele de date relaţionale
- ✓ Modele de reprezentare a bazelor de date relaţionale
- ✓ Arhitectura bazelor de date relaţionale
- ✓ Analiza, implementarea şi prelucrarea bazelor de date cu ajutorul sistemului de gestiune a bazelor de date **ORACLE DATABASE**

Laborator:

- ✓ Se vor cunoaşte toate elementele de utilizare a unui sistem de gestiune a bazelor de date **SQL** (Structured Query Language)
- ✓ Realizarea unor aplicatii de gestiune a bazelor de date in **APEX** (Application Express)

Introducere

- Pentru o bună înţelegere a noţiunilor teoretice şi practice prezentate în acest curs, este necesară parcurgerea anterioară a disciplinelor Programarea calculatoarelor, Proiectarea algoritmilor si Programare orientate pe obiecte.

- Cursul de **Baze de date** este structurat în 13 cursuri (capitole), fiecare dintre acestea cuprinzând câte un număr de 13 laboratoare (activităţi practice) la care prezenţa va fi obligatorie.

Conţinutul cursului

În cadrul acestui curs se vor studia bazele de date prelucrate cu ajutorul sistemului de gestiune a bazelor de date

ORACLE Database 12c Express Edition

Conţinutul cursului

Limbajul de interogare a bazelor de date în care se vor face exemplificările noţiunilor teoretice va fi **SQL** (Structured Query Language).

Capitolele cursului

1. **Sisteme de baze de date**. Modelul de date relaţional. Modelul de date orientate obiect.
2. **Evoluţia şi facilităţile sistemului ORACLE**. Arhitectura sistemului ORACLE. ORACLE SERVER. Oracle Database 11g Express Edition
3. **Limbajul SQL**. Introducere. Prezentare generală. Cereri SELECT pe o tabelă
4. **Cereri SELECT pe o tabelă**. Clauza WHERE. Clauza ORDER BY
5. **Funcţii**. Funcţii referitoare la o singură înregistrare
6. **Funcţii referitoare la mai multe înregistrări** (Funcţii de grup). Clauza GROUP BY. Excluderea grupurilor (clauza HAVING). Imbricarea funcţiilor de grup

Capitolele cursului

7. **SUBQUERIES** (Subinterogări). SINGLE ROW SUBQUERIES. MULTIPLE ROW SUBQUERIES
8. **Cereri din mai multe tabele** (JOIN-uri). JOIN-urile proprietatea ORACLE
9. **Cereri din mai multe tabele** (JOIN-uri). JOIN-urile ANSI/ISO SQL99. Operatorii pe mulţimi
10. **Limbajul de manipulare al datelor (**LMD). Tranzacţii(Transactions)
11. **Constrângeri**(Constrains)
12. **Vederi** (Views)
13. **Alte obiecte din baza de date**. Gestiunea utilizatorilor

Orice firmă utilizează baze de date pentru păstrarea şi gestionarea informaţiilor. Câteva astfel de aplicaţii sunt uzuale:

1.bazele de date ale liniilor aeriene care sunt accesate simultan din sute de agenţii pentru a realiza rezervări şi vânzări de locuri pentru date şi zboruri diferite

2.bazele de date ale băncilor care permit realizarea a mii de tranzacţii zilnic

3.bazele de date ale supermagazinelor care sunt accesate atât de la casele de marcaj cât şi de la echipamentele de inventariere

4.bazele de date ale bibliotecilor care păstrează milioane de titluri şi permit localizarea unei lucrări folosind diferite criterii (cuvinte cheie, titlu, autori, domeniu)

Pentru realizarea unei aplicaţii care foloseşte baze de date se poate proceda în două moduri:

a)Se creează baza de date cu ajutorul unei aplicaţii de tip server de baze de date şi se scriu apoi *aplicaţiile care accesează baza de date* într-un limbaj care posedă funcţiile necesare accesării server-ului (frecvent se folosesc limbajele C++, Java, C# sau Visual Basic)

b)Se foloseste o aplicaţie de tip sistem de gestiune de baze de date (S.G.B.D. sau D.B.M.S. - DataBase Management System).

Un astfel de sistem oferă un *ansamblu de instrumente software cu ajutorul cărora se crează atât baza de date cât şi aplicaţiile prin care aceasta este exploatată*.

Pentru utilizatorii sistemului de operare Windows cele mai cunoscute sisteme de acest fel sunt Access şi Visual FoxPro.

Noţiuni introductive despre teoria generală a bazelor de date

1.1. Sisteme de baze de date

1.2. Modelul de date relaţional

1.3. Modelul de date orientate obiect

1.4. Modelul de date obiect-relaţional

1.5. Modelul de date ierarhic

1.6. Modelul de date reţea

1.1.Sisteme de baze de date

Un **sistem de baze de date** este un sistem computerizat de evidenţă a informaţiilor.

Informaţia într-un sistem de baze de date consta atât din date cât şi din informaţii despre date (**metadate**) cum ar fi relaţiile dintre date.

1.1.Sisteme de baze de date

Un sistem de baze de date poate fi considerat ca având patru părţi:

1. date

2. utilizatori

3. hardware

4. software

1.1.Sisteme de baze de date

Datele:

Sunt informaţii pe care diferiţi utilizatori (firme, agenţii, sau simpli utilizatori) le colectează pentru a-şi îndeplini scopurile sau misiunile.

Datele individuale sunt stocate în mulţimi de date relaţionate (legate) numite **înregistrări**.

O colecţie de înregistrări dependente se numeşte **bază de date**.

1.1.Sisteme de baze de date

Utilizatorii:

Diferite persoane sau grupuri de persoane care folosesc informaţiile sunt definite ca utilizatori.

Hardware: De obicei noţiunea de hardware constă din device-uri fizice, cum ar fi harddisk-uri, imprimante, interfeţe de intrare/ieşire şi procesorul de date cu memoria sa asociată.

1.1.Sisteme de baze de date

Software: Interfaţa dintre datele fizice şi utilizatorul se numeşte **Sistemul de Gestiune a Bazelor de Date (SGBD)**.

SGBD-ul este un sistem software, dar poate conţine şi hardware specializat pentru a gestiona mai eficient datele.

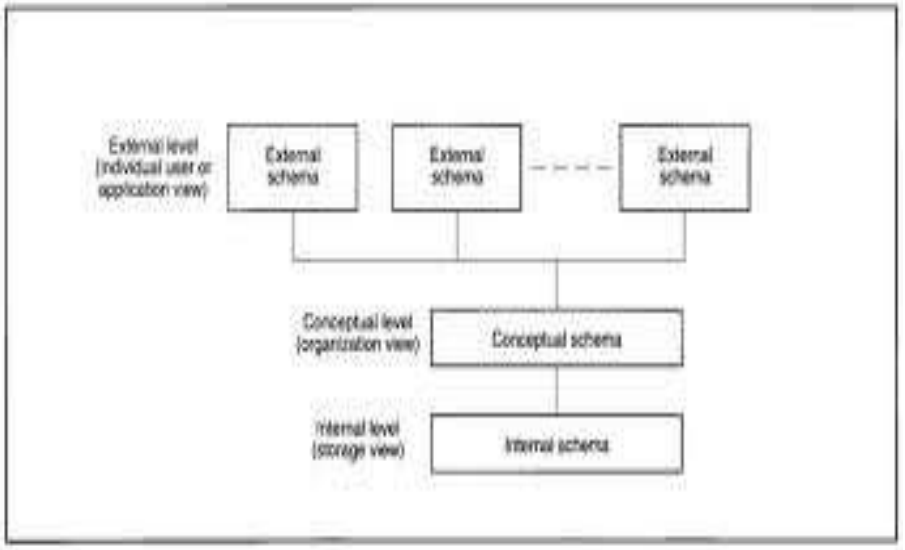
Aceste componente hardware pot fi harddisk-uri speciale care permit un acces mai rapid la date, sau multiprocesoare care permit procesarea paralelă de date.

1.1.Sisteme de baze de date

SGBD-ul furnizează diferiţilor utilizatori ai bazei de date, diferite modalităţi de lucru cu date în funcţie de necesităţile fiecăruia.

Aceste diferite modalităţi de lucru cu datele reprezintă diferite nivele de abstractizare al datelor:

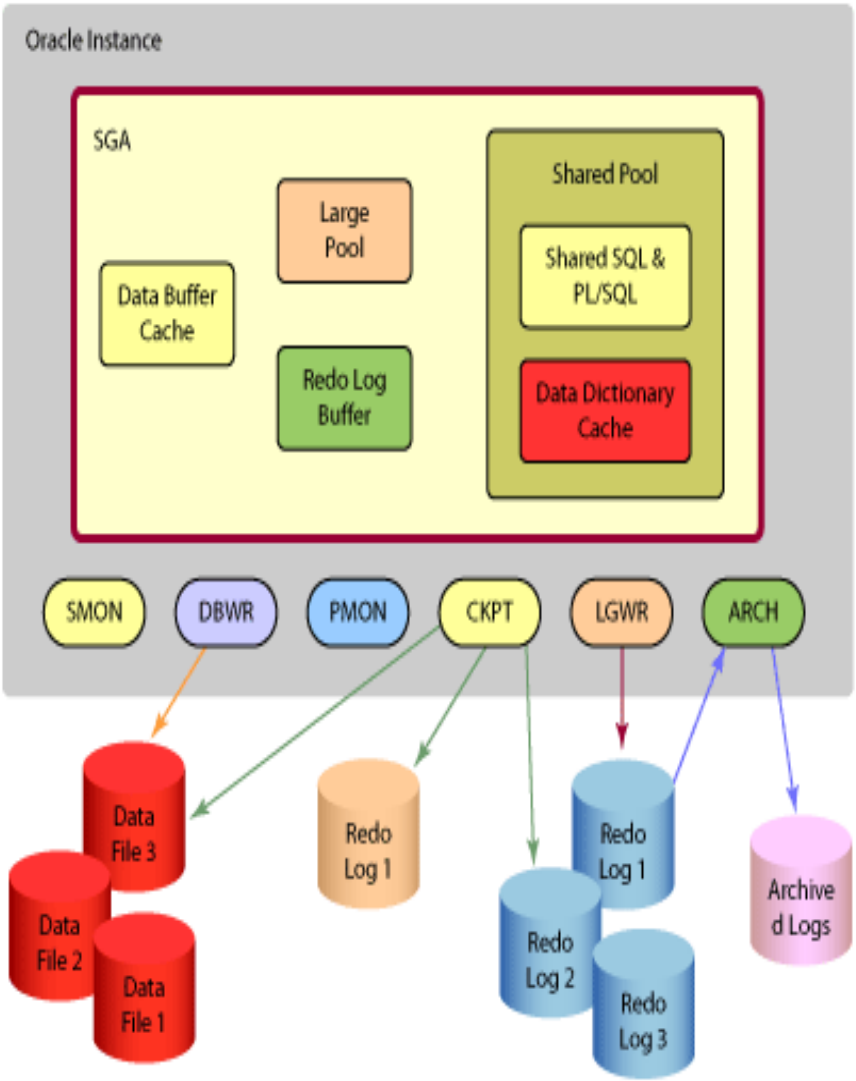
1. **Nivelul fizic**
2. **Nivelul conceptual**
3. **Nivelul de vizualizare**



1.1.Sisteme de baze de date

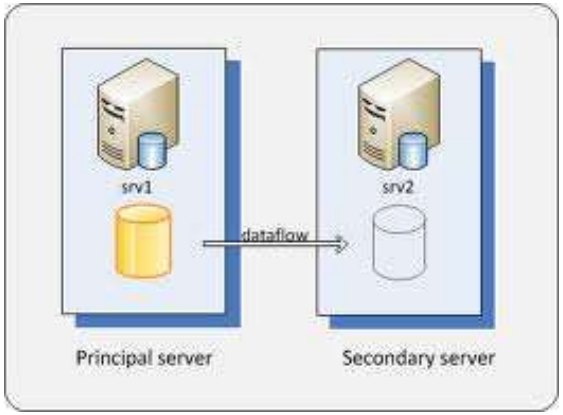
1. Nivelul fizic este cel mai de jos nivel de abstractizare.

De obicei, acest nivel este utilizat de programatorii SGBD-ului, care sunt interesaţi de cum anume se memorează datele pe suportul fizic.



1.1.Sisteme de baze de date

2. Nivelul conceptual este nivelul de mijloc al abstractizării, şi care se concentrează pe descrierea datelor care sunt în baza de date şi pe relaţiile dintre aceste date.



De acest nivel de abstractizare sunt interesaţi:

- *Administratorii bazei de date*
- *Administratorii securităţii bazelor de date*

1.1.Sisteme de baze de date

3. Nivelul de vizualizare este cel mai înalt nivel de abstractizare.

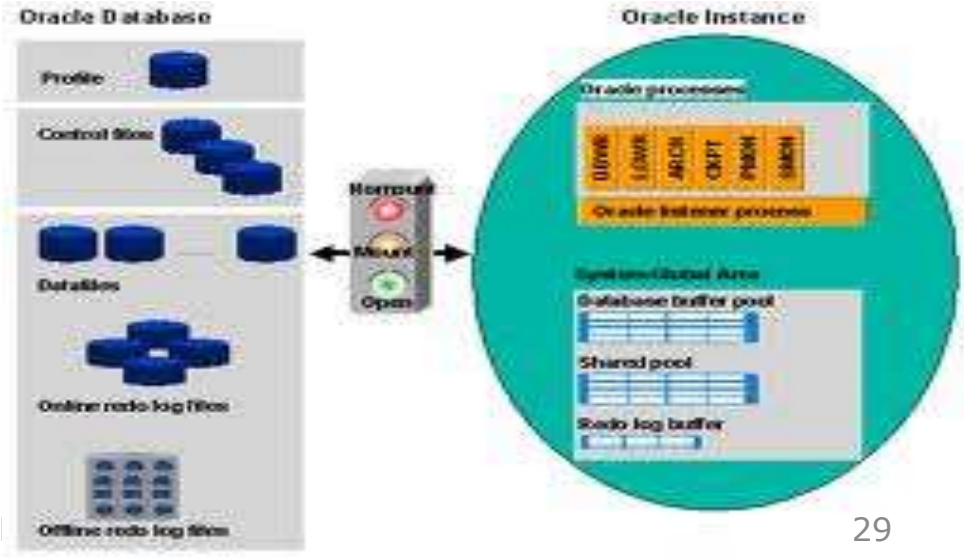
De obicei, acest nivel este modalitatea prin care utilizatorii finali folosesc datele.

Fiecare utilizator final are o vizualizare specifică asupra datelor pe care îl interezează.

Această vizualizare a datelor nu presupune ca utilizatorul să stie sau să înţeleagă caracteristicile interne ale datelor (cum ar modalitatea lor de reprezentare sau de stocare).

1.1.Sisteme de baze de date

Modelele bazelor de date permit diferenţierea dintre *descrierea bazei de date, care este specificată în schemă*, şi colecţia de conţinuturi sau de valori ale datelor din baza de date la un moment dat, care se numeşte **instanţă**.



1.1.Sisteme de baze de date

Schema bazei de date utilizează un **limbaj de definire a datelor (DDL – Data Definition Language)**.

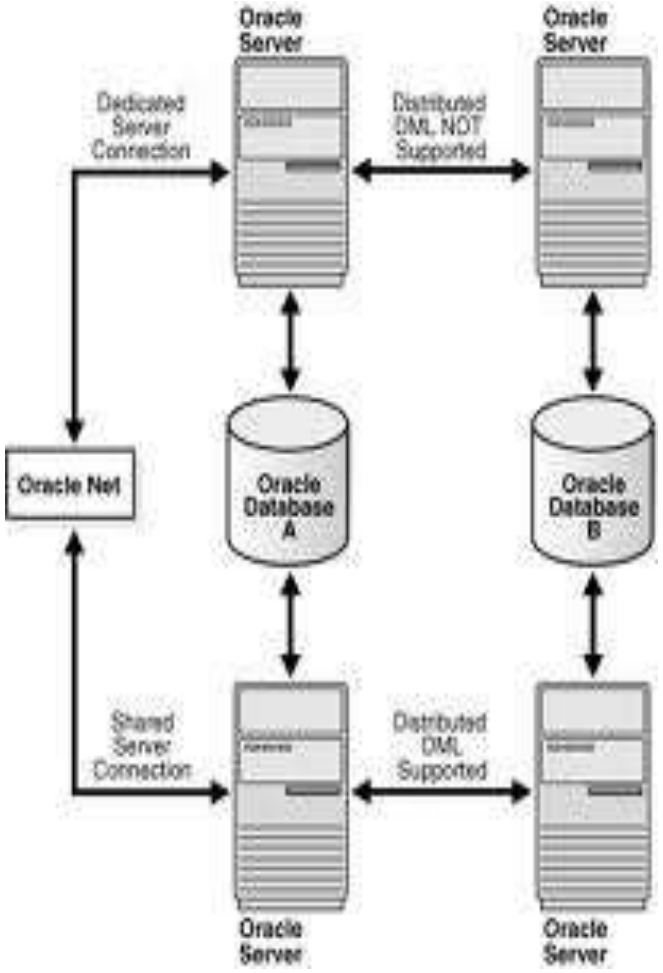
Manipularea datelor în baza de date (inserare, ştergere, actualizare, sau recuperare de valori de date) se poate face cu ajutorul **limbajului de manipulare datelor (DML – Data Manipulation Language)**.

1.1.Sisteme de baze de date

SGBD-ul utilizează o mulţime complexă de componente software pentru a-şi îndeplini funcţiile sale.

Aceste componente includ:

- **managerul de date** care furnizează o interfaţă către datele stocate fizic în baza de date;
- **procesorul de interogări** care traduce limbajul de interogare în instrucţiuni pentru managerul de date;



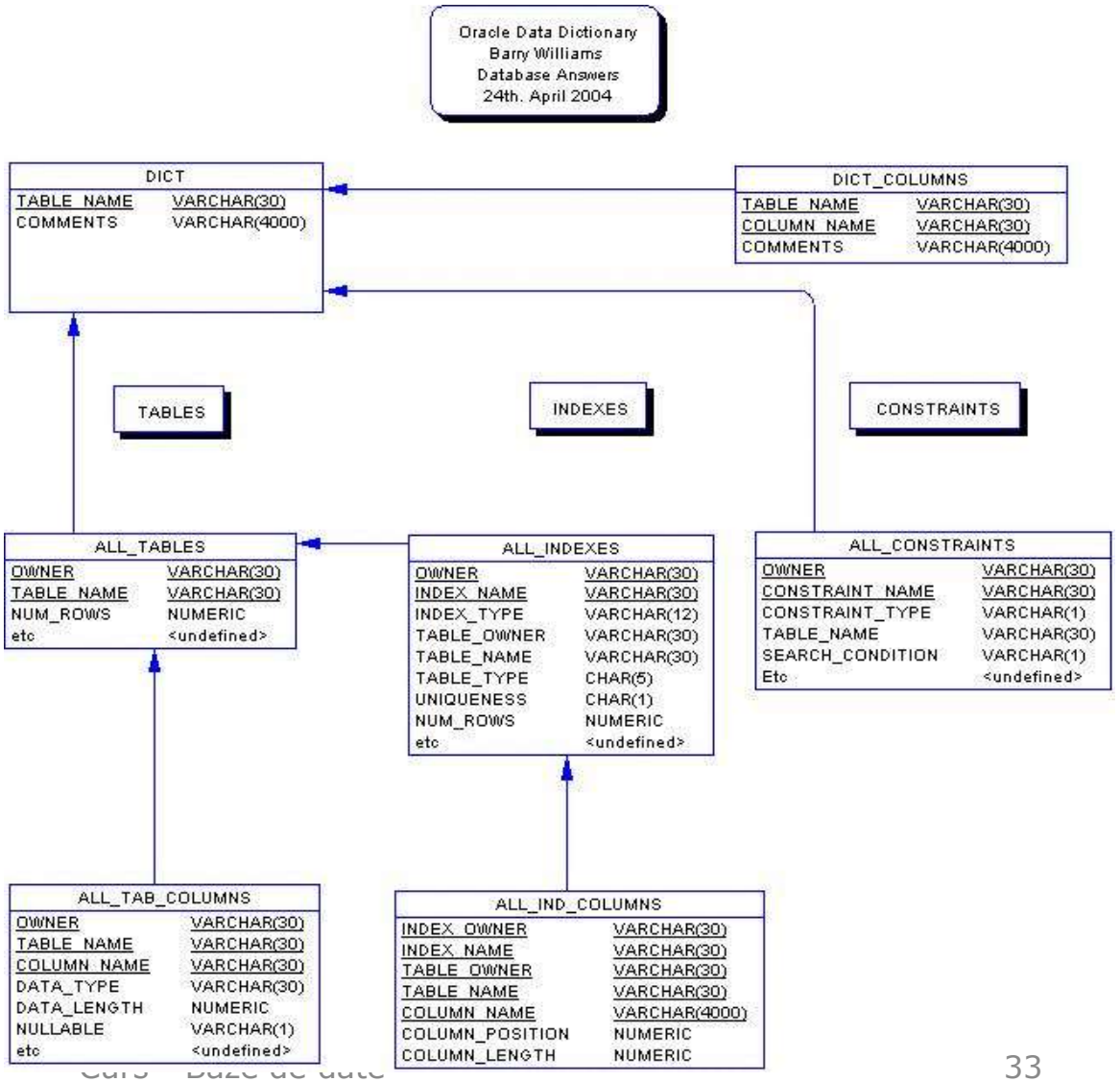
1.1.Sisteme de baze de date

- **precompilatorul limbajului de manipulare a datelor** care transformă instrucţiunile **DML** de la programele de aplicaţii la limbajul gazdă, şi lucrează cu procesorul de interogări;
- **compilatorul limbajului de definiţie a datelor** care transformă instrucţiunile **DDL** în tabele de metadate.

1.1.Sisteme de baze de date

Metadatele sunt memorate în **dicţionarul de date**, care include:

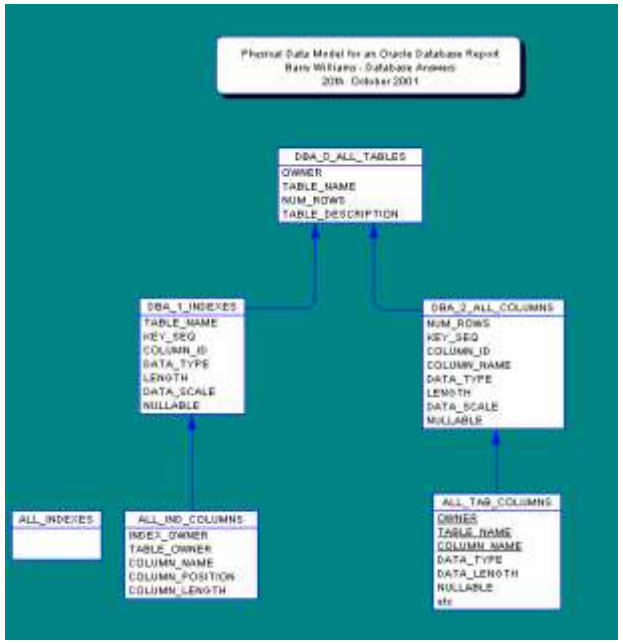
- **structura bazei de date** sau **schema**
- **constrângerile de integritate**
- **constrângerile de securitate**



Modele de baze de date

Vom prezenta acum două din modele de baze de date care sunt cele mai utilizate:

1. Modelul de date relaţional



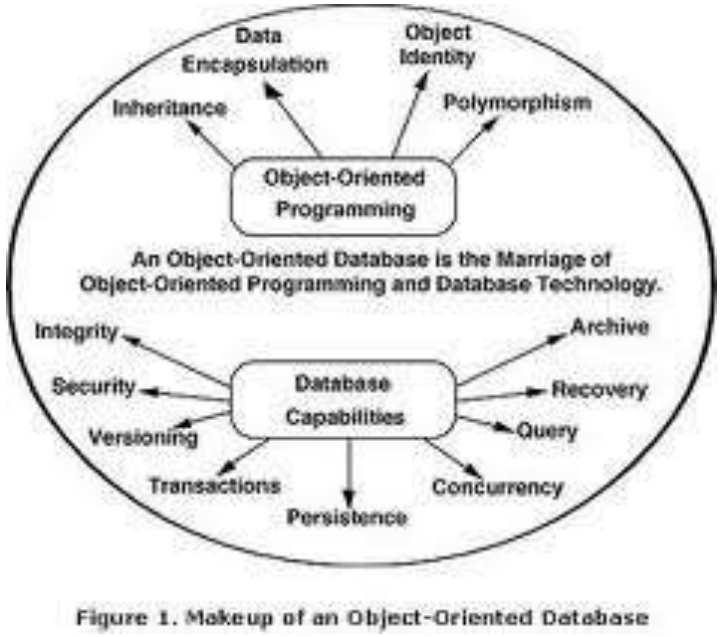


Figure 1. Makeup of an Object-Oriented Database

2. Modelul de date orientate obiect

Noţiuni introductive despre teoria generală a bazelor de date

1.1. Sisteme de baze de date

1.2. Modelul de date relaţional

1.3. Modelul de date orientate obiect

1.4. Modelul de date obiect-relaţional

1.5. Modelul de date ierarhic

1.6. Modelul de date reţea

1.2.Modelul de date relaţional

O bază de date este reprezentată cu ajutorul **modelului relaţional** ca o colecţie de tabele.

Mult mai important, este faptul că acest model este direct legat de conceptul matematic de relaţie şi este compus din:

1. **o parte structurală.** Schema bazei de date este o colecţie de scheme de relaţii şi o bază de date este o colecţie de relaţii.
2. **o parte de integritate.** Chei primare şi chei străine.
3. **o parte de manipulare.** Algebra relaţională şi calcul relaţional.

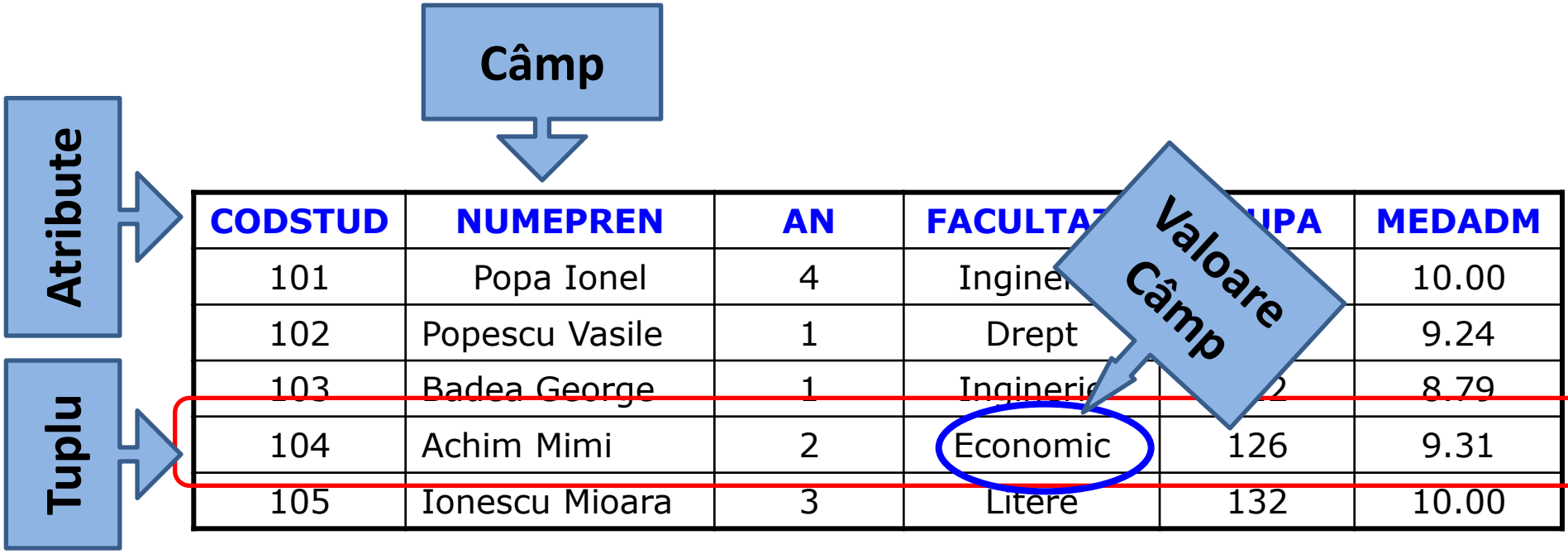
1.2.Modelul de date relaţional

Formal, o relaţie **R** este o mulţime, unde $\mathbf{D_1}$, $\mathbf{D_2}$, ..., $\mathbf{D_n}$ sunt **domeniile** a **n atribute** $\mathbf{A_1}$, $\mathbf{A_2}$, ..., $\mathbf{A_n}$.

- ✓ Elementele relaţiei sunt **n**-tuplurile ($\mathbf{v_1}$, $\mathbf{v_2}$, ..., $\mathbf{v_n}$) cu $\mathbf{v_i}$ $\mathbf{D_i}$, adică valoarea celui de-al i-lea atribut trebuie să fie un element din mulţimea $\mathbf{D_i}$.
- ✓ Elementele dintr-un tuplu sunt numite **câmpuri**.
- ✓ Când un camp nu conţine nici o valoare, vom reprezenta acest fapt cu o valoare specială numită **valoare null**, semnificând că "nu este nici o intrare" în loc de "intrarea este necunoscută".

1.2.Modelul de date relaţional

Prezentăm în continuare un exemplu de tabelă pentru gestiunea studenţilor dintr-o facultate:



CODSTUD	NUMEPREN	AN	FACULTA	UPA	MEDADM
101	Popa Ionel	4	Inginer		10.00
102	Popescu Vasile	1	Drept		9.24
103	Badea George	1	Ingineri	2	8.79
104	Achim Mimi	2	Economic	126	9.31
105	Ionescu Mioara	3	Litere	132	10.00

1.2.Modelul de date relaţional

Constrângeri de integritate

Constrângerile de integritate restricţionează mulţimea tuplu-rilor teoretice posibile la o mulţime care este în mod practic cu o semnificaţie.

Fie X şi Y două mulţimi cu unul sau mai multe atribute A_i, din schema relaţională.

Spunem că Y este dependent funcţional de X, şi notăm acest lucru prin $X \rightarrow Y$, dacă şi numai dacă nu este posibil să avem două tupluri cu aceeaşi valoare pentru toate atributele din X dar cu valoare diferită pentru toate atributele din Y.

1.2.Modelul de date relaţional

Cele mai importante constrângeri de integritate sunt:

Constrângerea de integritate a entităţii care stabileşte că fiecare tuplu să fie unic identificat printr-o cheie şi atributul cheie să nu poată fi null;

CODSTUD	NUMEPREN	AN	FACULTATE	GRUPA	MEDADM
101	Popa Ionel	4	Inginerie	145	10.00
102	Popescu Vasile	1	Drept	113	9.24
103	Badea George	1	Inginerie	112	8.79
104	Achim Mimi	2	Economic	126	9.31
105	Ionescu Mioara	3	Litere	132	10.00

1.2.Modelul de date relaţional

Constrângerea referenţială de integritate stabileşte că un n-tuplu dintr-o relaţie care se referă la o altă relaţie, trebuie să se refere la un n-tuplu care există în acea relaţie; această condiţie se referă la cheile străine.

1.2.Modelul de date relaţional

O **cheie candidat** a unei relaţii **R** este o mulţime minimă de atribute de care toate celelalte atribute ale lui **R** sunt dependente funcţional.

Cheia primară a unei relaţii **R** este una din cheile candidat care a fost desemnată în acest scop.

O cheie străină a unei relaţii **R** este o mulţime de atribute din schema relaţională care formează o cheie primară pentru o altă relaţie.

CODSTUD	NUMEPREN	AN	FACULTATE	GRUPA	MEDADM
101	Popa Ionel	4	Inginerie	145	10.00
102	Popescu Vasile	1	Drept	113	9.24
103	Badea George	1	Inginerie	112	8.79
104	Achim Mimi	2	Economic	126	9.31
105	Ionescu Mioara	3	Litere	132	10.00

Noţiuni introductive despre teoria generală a bazelor de date

1.1. Sisteme de baze de date

1.2. Modelul de date relaţional

1.3. Modelul de date orientate obiect

1.4. Modelul de date obiect-relaţional

1.5. Modelul de date ierarhic

1.6. Modelul de date reţea

1.3.Modelul de date orientate obiect

Bazele de date orientate obiect permit crearea unor obiecte complexe din componente mai simple, fiecare având atribute proprii şi comportament specific.

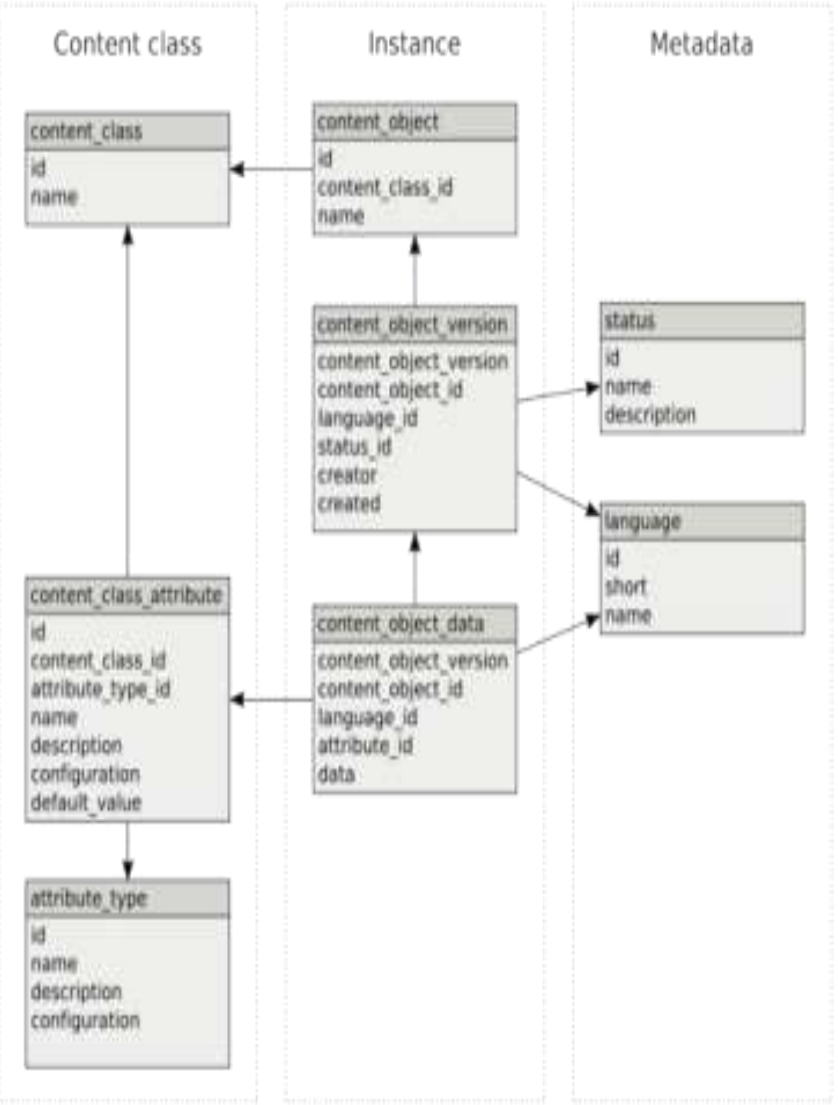
Aceste sisteme combină posibilitatea definirii şi manipulării structurilor complexe de date cu funcţionalitatea unui limbaj de programare şi tehnologia de gestiune a bazelor de date.

1.3.Modelul de date orientate obiect

Modelele de date orientate obiect (MDOO) au fost create pentru a modela lumea reală.

De exemplu, limbajul C++ a fost dezvoltat pentru a crea modelul unui sistem telefonic.

Conceptul fundamental al unui MDOO este obiectul.



1.3.Modelul de date orientate obiect

Într-un **MDOO**, orice entitate din lumea reală este un obiect şi reciproc, orice obiect reprezintă o entitate a lumii reale.

Un ***obiect*** reprezintă un grup de date stucturate, identificate printr-o referinţă unică.

Un ***obiect persistent*** este un obiect stocat în baza de date care are o durată de viaţă mai mare decât a programului care l-a creat.

Un ***obiect tranzitoriu*** este un obiect depus în memorie, a cărui durată de viaţă nu depăşeşte durata de viaţă a programului care l-a creat.

1.3.Modelul de date orientate obiect

Baza de date orientate obiect (BDOO) este o organizare coerentă de obiecte persistente, partajate de utilizatori concurenţi.

Prin urmare, **BDOO** este rezultatul aplicării tehnologiei orientate obiect în domeniul stocării şi găsirii informaţiilor.

Schema unei **BDOO** trebuie să includă:

- definiţiile structurale (atribute şi tipuri)
- definiţiile comportamentale (metode) ale obiectelor

1.3.Modelul de date orientate obiect

Un **sistem de gestiune al unei baze de date orientate obiect (SGBDOO)** trebuie să îndeplinească cerinţele unui SGBD şi să fie în plus, un sistem orientat pe obiecte.

Aceste două criterii generează o mulţime de caracteristici ale unui SGBDOO.

1.3.Modelul de date orientate obiect

Putem accepta ca definiţie minimală:

SGBDOO = SGBD + obiect + moştenire + polimorfism

1.3.Modelul de date orientate obiect

Caracteristicile obligatorii ale unui SGBDOO sunt:

1. ***Manipularea obiectelor atomice şi complexe*** (colecţii imbricate).

Un constructor este o funcţie asociată unei clase care permite crearea şi iniţializarea unui obiect (în memorie).

Un destructor este o funcţie asociată unei clase care permite distrugerea unui obiect.

Noţiunea de obiect complex s-a născut prin aplicarea de constructori asupra obiectelor simple.

O condiţie privind constructorii, referitoare la **MDOO**, o constituie *ortogonalitatea* care presupune ca fiecare constructor să fie aplicabil fiecărui obiect.

1.3.Modelul de date orientate obiect

2. Persistenţa obiectelor. Obiectele pot persista mai mult decât programul care a creat aceste obiecte.

3. Concurenţa acceselor. BDOO poate să fie partajată simultan de către tranzacţiile care o consultă şi o modifică.

4. Fiabilitatea obiectelor. În cazul unei defecţiuni, obiectele trebuie restaurate la starea pe care au avut-o înainte de defecţiune.

1.3.Modelul de date orientate obiect

5. Uşurinţa interogării.

Un obiect poate fi găsit utilizând valorile atributelor sale, legăturile cu alte obiecte sau metodele aplicate acestuia.

6. Identitatea obiectelor.

Orice obiect trebuie să aibă un identificator sistem.

1.3.Modelul de date orientate obiect

7. Moştenirea (simplă).

O clasă poate fi specializarea altei clase şi, prin urmare, poate să o moştenească.

Moştenirea reduce efortul de programare.

Există mai multe modalităţi de a moşteni şi anume prin:

1. substituţie
2. incluziune
3. restricţie
4. specializare

1.3.Modelul de date orientate obiect

8. Polimorfismul.

Codul unei metode trebuie ales în funcţie de parametrii săi.

9. Extensibilitatea.

SGBDOO trebuie să includă pe lângă clasele sale şi tipurile predefinite şi instrumentele care să permită utilizatorului definirea unor noi clase şi tipuri.

1.3.Modelul de date orientate obiect

Dintre caracteristicile opţionale ale unui SGBDOO amintim:

- ✓ ***Distribuţia obiectelor.*** Această distribuţie permite gestionarea obiectelor în diferite staţii.
- ✓ ***Modelarea tranzacţiilor evoluate.*** Ideea este de a accepta tranzacţii imbricate care pot fi descompuse în subtranzacţii.
- ✓ ***Versiuni ale obiectelor.*** Plecând de la un anumit obiect, prin modificări succesive sau paralele, pot fi obţinute mai multe versiuni ale obiectului.

1.3.Modelul de date orientate obiect

- ***Moştenirea multiplă.*** O clasă (subclasă) poate fi specializarea directă a unor supraclase şi să moştenească proprietăţile acestora.

- ***Mesajele de eroare.*** Este vorba de un mecanism de detectare şi tratare a erorilor care implică faptul că dacă într-o metodă apare o eroare, este trimis un mesaj unei clase speciale definită anterior, care o va înregistra şi o va trata corespunzător.

Noţiuni introductive despre teoria generală a bazelor de date

1.1. Sisteme de baze de date
1.2. Modelul de date relaţional
1.3. Modelul de date orientate obiect
1.4. Modelul de date obiect-relaţional
1.5. Modelul de date ierarhic
1.6. Modelul de date reţea

1.4. Modelul de date obiect-relaţional

Modelul de date obiect-relaţional (Object-Relational Model) reprezintă extinderea modelului relaţional cu caracteristici ale modelului obiect, extindere necesară pentru realizarea bazelor de date care definesc şi prelucrează tipuri de date complexe.

1.4. Modelul de date obiect-relaţional

✓ În esenţă, modelul obiect-relaţional păstrează structurarea datelor în relaţii (reprezentate ca tabele), dar adaugă *posibilitatea definirii unor noi tipuri de date*, pentru domeniile de valori ale atributelor.

✓*Tipurile de date definite de utilizator pot fi extinse prin mecanismul de moştenire* şi pentru fiecare tip sau subtip se pot defini metode pe care le pot executa obiectele de acel tip.

Noţiuni introductive
despre teoria generală a bazelor de date

1.1. Sisteme de baze de date
1.2. Modelul de date relaţional
1.3. Modelul de date orientate obiect
1.4. Modelul de date obiect-relaţional
1.5. Modelul de date ierarhic
1.6. Modelul de date reţea

1.5. Modelul de date ierarhic

- ✓ În modelul de date ierarhic (Hierarchical Model) o bază de date se reprezintă printr-o *structură ierarhică de înregistrări de date (records) conectate prin legături (links).*

- ✓ Modelul ierarhic a fost primul model folosit pentru dezvoltatea bazelor de date.

- ✓ Schema conceptuală a unei baze de date în modelul ierarhic se reprezintă printr-un număr oarecare de scheme ierarhice.

1.5. Modelul de date ierarhic

✓O schemă ierarhică este un *arbore direcţionat, reprezentat pe mai multe niveluri*, în care nodurile sunt tipurile de înregistări, iar *arcele sunt tipurile de legături*.

✓Fiecare nod (cu excepţia nodului rădăcină) are o singură legătură către un nod de pe un nivel superior (nodul părinte) şi fiecare nod (cu excepţia nodurilor frunză) are una sau mai multe legături către noduri de pe nivelul imediat inferior (noduri fii).

Noţiuni introductive despre teoria generală a bazelor de date

1.1. Sisteme de baze de date
1.2. Modelul de date relaţional
1.3. Modelul de date orientate obiect
1.4. Modelul de date obiect-relaţional
1.5. Modelul de date ierarhic
1.6. Modelul de date reţea

1.6. Modelul de date reţea

Modelul de date reţea (Network Model) foloseşte o structură de graf pentru definirea schemei conceptuale a bazei de date:

- nodurile grafului sunt tipuri de entităţi (înregistrări, records),
- iar muchiile grafului reprezintă în mod explicit asocierile (legăturile, links) dintre tipurile de entităţi.

1.6. Modelul de date reţea

- ✓ La fel ca şi modelul ierarhic, dezavantajul principal al modelului reţea este acela că *fiecare interogare trebuie să fie prevazută încă din faza de proiectare*, prin memorarea explicită a legăturilor între tipurile de entităţi.

- ✓ În plus, complexitatea reprezentării datelor în modelul reţea este deosebit de ridicată, iar programatorii trebuie să o cunoască pentru a putea realiza aplicaţiile necesare.

Sisteme de baze de date

Alte clasificari ale sistemelor de baze de date:

1. Clasificare după numărul de utilizatori

2. Clasificare după numărul de staţii pe care este stocată baza de date

1. Clasificare dupa numărul de utilizatori

1. Majoritatea sistemelor de baze de date sunt sisteme multiutilizator, adică permit accesul concurent (în acelaţi timp) a mai multor utilizatori la aceeaşi bază de date.

2. Există şi un număr redus de sisteme monoutilizator, adică suportă accesul doar al unui utilizator (la un moment dat).

2. Clasificare după numărul de staţii pe care este stocată baza de date

✓ Există două categorii de sisteme de baze de date:

1. centralizate

2. distribuite

1. Un sistem de baze de date centralizat (Centralized Database System) este un sistem de baze de date în care datele şi sistemul de gestiune sunt stocate pe un singur calculator.
2. Un sistem de baze de date distribuit (Distributed Database System) poate avea atât datele, cât şi sistemul de gestiune, distribuite pe mai multe calculatoare interconectate printr-o reţea de comunicaţie.

Întrebări?